CLAIMS

What is claimed is:

1. For an electronic architecture with a functional constitution, said functional constitution performing a number of architectural functions characterized by separate stages, a method of deriving a benchmark program for maximum power consumption prior to an implementation of said architecture, said method comprising:

modeling a functional model of said architecture;

compiling said benchmark program into a corresponding instruction stream;

valuating a power weight for each said stage of each said function of each said constituent;

running said model in a maximum power consumption mode; and

summarizing said power consumption.

2. The method as recited in Claim 1, wherein said functional model comprises a representation of said functional constitution of said architecture at a high level of abstraction capable of simulating the functioning of said architecture and said functional constitution thereof, the total power consumption of said architecture and of said functional constitution thereof, wherein said benchmark delineates power consumption of said is in terms of maximum power.

3. The method as recited in Claim 2, wherein said modeling a functional model of said architecture further comprises writing a program in SystemC.

4. The method as recited in Claim 1, wherein said compiling said benchmark program into a corresponding instruction stream is performed by a compiler.

5. The method as recited in Claim 4, wherein said compiler is a PERL script.

6. The method as recited in Claim 1, wherein said valuating a power weight for each said stage of each said function of each said constituent further comprises:

selecting each of said architectural functions individually;

determining the characteristic technology of each of said architectural functions selected;

counting a number of technology gates constituting each of said architectural functions selected;

determining a power weight for each of said technology gates; and

deriving a power weight for each of said architectural functions selected.

7. The method as recited in Claim 6, wherein said selecting each of said architectural functions individually further comprises:

determining if said architectural functions selected are memory type functions;

itemizing individual memory subfunctions; and

treating each of said individual memory subfunctions as separate, equivalent, distinct architectural functions.

8. The method as recited in Claim 6 wherein said deriving a power weight for each of said architectural functions selected comprises multiplying said power weight determined for each of said technology gates by said number of said technology gates.

9. The method as recited in Claim 1, wherein said running said model in a maximum power consumption mode comprises running a power virus program.

10. The method as recited in Claim 1, wherein said architecture is a microprocessor.

11. For an electronic architecture with a functional constitution, said functional constitution performing a number of architectural functions characterized by separate stages performing a constituent subfunction, a method of utilizing a benchmark for maximum power consumption by said architecture and said architectural functions to analyze design corresponding to said architecture, said method comprising:

determining said benchmark for power consumption by said architecture and said architectural functions;

specifying a design analysis to be performed;

selecting one of said functions;

designating one of said stages constituting said one of said functions selected;

removing a set of instruction set architecture instructions corresponding to said one of said stages designated;

emulating said constituent subfunction corresponding to said one of said stages designated; and

summarizing a resulting operation into a summary.

12. The method as recited in Claim 11, wherein said determining said benchmark for power consumption by said architecture and said architectural functions further comprises:

modeling a functional model of said architecture;

compiling said benchmark into a corresponding instruction stream;

valuating a power weight for each said stage of each said function of each said constituent;

inserting said power weight into said instruction stream at a corresponding appropriate place therein;

running said model in a maximum power consumption mode; and

summarizing said power consumption.

13. The method as recited in Claim 12, wherein said modeling a functional model of said architecture further comprises writing a program in SystemC.

14. The method as recited in Claim 12, wherein said compiling said benchmark into a corresponding instruction stream is performed by a compiler.

15. The method as recited in Claim 14, wherein said compiler is a PERL script.

16. The method as recited in step 12, wherein said valuating a power weight for each said stage of each said function of each said constituent further comprises:

selecting each of said architectural functions individually;

determining the characteristic technology of each of said architectural functions selected;

counting a number of technology gates constituting each of said architectural functions selected;

determining a power weight for each of said technology gates; and

deriving a power weight for each of said architectural functions selected.

17. The method as recited in Claim 12, wherein said running said model in a maximum power consumption mode further comprises running a power virus program.

18. The method as recited in Claim 11, wherein said emulating said constituent subfunction corresponding to said one of said stages designated further comprises:

generating a summary of power consumption results for said architecture and said constituent functions; and

generating a summary of performance for said architecture.

19. The method as recited in Claim 11, further comprising the step of:

analyzing said summary.

20. The method as recited in Claim 19, wherein said analyzing said summary further comprises balancing power consumption and performance.

21. The method as recited in Claim 11, wherein said architecture is a microprocessor.

22. The method as recited in Claim 11, wherein said architecture is an integrated circuit.

23. The method as recited in Claim 22, wherein said architecture is an application specific integrated circuit.